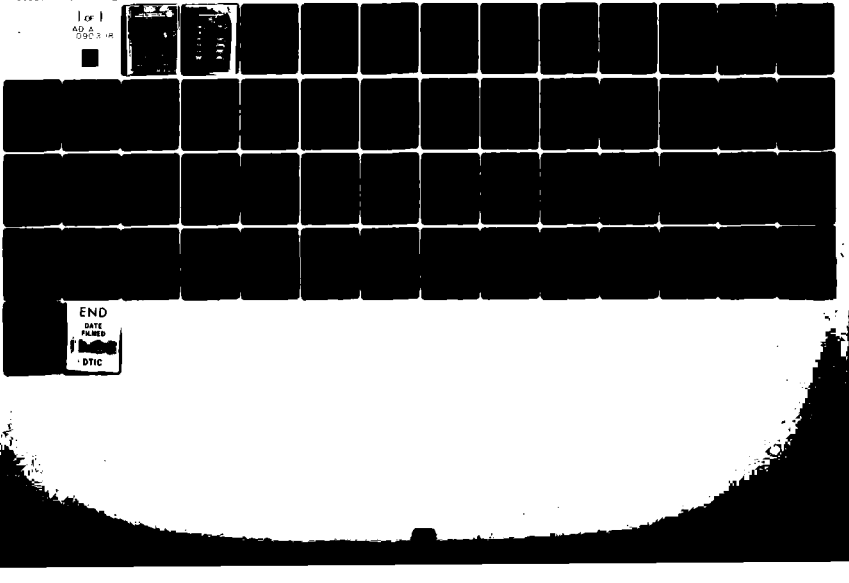
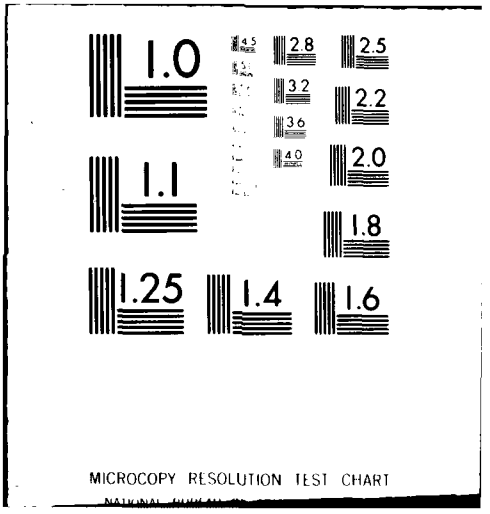
AD-A090 308 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
CONTAINER ORIENTED NETWORK DISTRIBUTION ANALYSIS (CONDA) SYSTEM--ETC(U)
SEP 80 F A MYERS, R B WILSON

UNCLASSIFIED DTNSRDC-80/1118 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
MICROCOPY RESOLUTION TEST CHART

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: DTNSRDC-80/118 (14)	2. GOVT ACCESSION NO.: AD-A090308
	3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): CONTAINER ORIENTED NETWORK DISTRIBUTION ANALYSIS (CONDA) SYSTEM.	5. TYPE OF REPORT & PERIOD COVERED: Final
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Fred A. Myers, Ronald B. Wilson	8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Program Element 62760N, Task Area TF 60 531 091, Work Unit 1800-007
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Supply Systems Command/043, Research and Development Division, Washington, D.C. 20390	12. REPORT DATE: September 1980
	13. NUMBER OF PAGES: 53
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Distribution Analysis
Transportation Systems
Data Base Management Systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Container Oriented Network Distribution Analysis (CONDA) Program was developed to provide the Navy with the capability to determine the extent, types, and manner in which containers/unitized cargo should be handled and incorporated into the Navy Supply System. The CONDA program encompassed two complementary efforts. The initial effort resulted in

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

EC/622

(Block 20 continued)

the design and development of a data base which addresses the specific areas of inland cargo movement, port handling, and ocean freight container movement. This prototype data base operates with the SHARP Data Base Management System on a Control Data Corporation 6700 computer at the David W. Taylor Naval Ship Research and Development Center.

The second major effort of the CONDA Program is a study, conducted by J.J. Henry Co., Inc., that addresses the containerized cargo and documentation flow for seavan cargo that moves from origin to destination through a water terminal.

The CONDA Program provides management with a decision making tool, supported by actual historical data, to analyze and evaluate route patterns and rate structures for both break-bulk and seavan cargo that moves throughout the Navy Supply System.

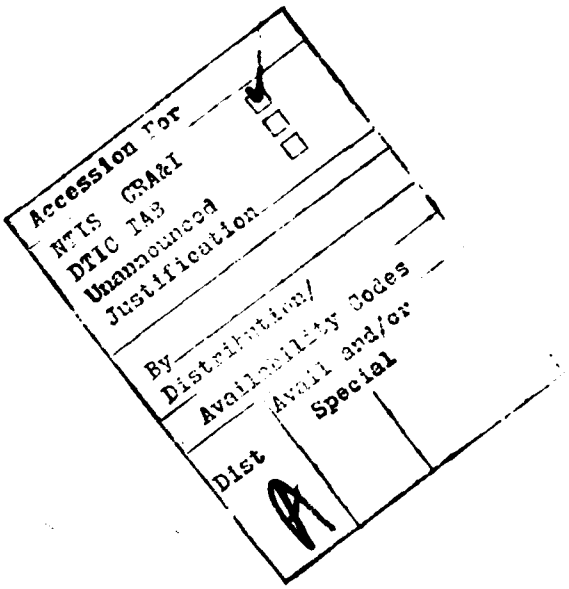


TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Container Oriented Network Distribution Analysis (CONDA) Program was developed to provide the Navy with the capability to determine the extent, types, and manner in which containers/unitized cargo should be handled and incorporated into the Navy Supply System. The CONDA program encompassed two complementary efforts. The initial effort resulted in the design and development of a data base which addresses the specific areas of inland cargo movement, port handling, and ocean freight container movement. This prototype data base operates with the SHARP Data Base Management System on a Control Data Corporation 6700 computer at the David W. Taylor Naval Ship Research and Development Center.

The second major effort of the CONDA Program is a study, conducted by J.J. Henry Co., Inc., that addresses the containerized cargo and documentation flow for seavan cargo that moves from origin to destination through a water terminal.

The CONDA Program provides management with a decision making tool, supported by actual historical data, to analyze and evaluate route patterns and rate structures for both break-bulk and seavan cargo that moves throughout the Navy Supply System.

ADMINISTRATIVE INFORMATION

The CONDA program was sponsored by NAVSUP 043 under program element number 62760N. NAVMAT 08D1 was the program element manager. DTNSRDC, Code 187, was the technical agent with support from J.J. Henry Co., Inc.

SECTION 1
INTRODUCTION

Since the mid 1960's commercial shipping has been steadily shifting to the use of container ships. By 1985 as much as 85 percent of U.S. flag sealift capacity may be in container capable ships. As the responsible agent for all DoD sealift and as a primary user of containerized cargo, the Navy must develop the most effective methods for containerized cargo utilization.

One of the Navy initiatives in this area is the development of the Container Oriented Network Distribution Analysis (CONDA) system. The primary objective of the CONDA system is to determine the best types of containers, the extent to which they should be used, and the manner in which they should be handled and incorporated into the Navy Supply System. The CONDA system provides management with a decision making tool, supported by actual historical data, for analyzing and evaluating route patterns and rate structures for both breakbulk and seavan cargo moving through the Navy Supply System. The specific areas that CONDA addresses include inland cargo movements, port handling, and ocean freight container movement.

CONDA, developed by Code 187 of the David W. Taylor Naval Ship Research and Development Center (DTNSRDC), consists of a data base management system, and a prototype data base, both of which are operational on the CDC 6700 computer System at DTNSRDC. The data base was established under the SHARP data base management system which allows for the creation of large files and the organization, storage, retrieval, maintenance, and security of the data elements within these files. The current prototype data base includes files of GBL payment data, port handling billing data, seavan cargo movement data, commercial container agreement data, and containerized contract shipment data. Presently the CONDA data base contains 83 data elements taken from these five data files. Future applications of CONDA may require additional data elements from other files. The data base could be easily updated by a user familiar with the SHARP Data Management System.

The CONDA program is designed for operation by management users with limited computer experience. It is envisioned as a management tool for use in the evaluation of route patterns and rate structures in origin-destination cargo movement.

Some of the potential uses to the Navy of the CONDA system capability include

- o analyzing and evaluating the Navy Supply System's cargo movement
- o independently assessing transportation costs
- o making tradeoffs of routes versus rates
- o determining alternatives for consolidating containers at terminals
- o checking all billing shipments by GBL or TCN number
- o determining container cube utilization
- o forecasting future operations.

A study conducted by J.J. Henry Co., Inc., demonstrated some of these potential uses. The study [1*] addresses the containerized cargo and documentation flow for seavan cargo moving from origin to destination through a water terminal in the Naval Supply System. This study analyzed shipments that move from the shipper through a Port of Embarkation (POE) to a Port of Discharge (POD), and finally reach an ultimate consignee. The study included both the stuffing and stripping of containers at the water terminal and shipments of source-stuffed seavans which move directly from vendor sources to the ultimate consignee. This study made use of many of the data elements collected as part of the prototype CONDA data base.

*A complete listing of references is given on page 47.

SECTION 2
THE CONDA DATA BASE

The CONDA data base is stored in the CDC 6700 computer located at the David W. Taylor Naval Ship Research and Development Center and can be accessed from remote user terminals by using an interactive query language.

2.1 CONDA DATA FILES

The CONDA system development began with the identification of the data elements involved in the movement of Navy cargo. These data elements represent cargo characteristics, transportation information, shipment unit information, and associated costs and were obtained from five separate sources:

- the Military Traffic Management Command (MTMC) Cargo Master Prime File (606019F)
- the Navy Material Transportation Office (NAVMTO) GBL Payment Data File
- the MTMC Port Handling File (HRSA11)
- the Military Sealift Command (MSC) Cost and Accrual File (F3902552)
- the DoD Activity Address Directory (DoDAAD) Master Record File

These five data files could not be linked through a common data element because transportation data are identified by transportation control numbers (TCN) and payment data are identified by Government Bill of Lading (GBL) numbers. Data elements selected from each of the files provide the information required for each segment of the transportation routes.

2.2 CONDA DATA BASE DEFINITION

The CONDA data base was designed under the SHARP Data Management System (DMS).[2] SHARP requires that certain information be provided for each data element that appears in the data base. This information is used to control the processing of data throughout the system.

2.2.1 Data Element Number

A unique data element number must be assigned to each data element to be entered into the CONDA data base. The data element number can be any value from 3 through 999. The numbers 0, 1, and 2 are unique SHARP identifier codes. See the SHARP User's Manual[2] for an explanation of these codes.

2.2.2 Data Element Name

The SHARP DMS requires that each data element in the data base be given a unique name. The acronym can be any combination of characters A through Z, numbers 0 through 9, and the special characters slash (/) and dash (-). Each acronym must be limited to a maximum of ten characters with no embedded blanks. Acronyms ending in the suffix "DATE" are assumed to identify a numeric field containing a date formatted YYMMDD or Julian YYDDD.

2.2.3 Data Element Size

Each data element must be identified as having either a fixed or a variable length data field by using the keywords SIZE or MAXSIZE, respectively. The size specification indicates the length of a data element and is required for all data elements. SHARP ensures that the length of the data values in a file maintenance transaction matches the length specified originally. If a difference is detected, the transaction is rejected as invalid.

2.2.4 Format

The CONDA data base will accommodate any type of data: alphanumeric, alphabetic, numeric, real, integer, or free format. However, the format for each data element must be specified. In file maintenance transactions SHARP will test data values for the correct format.

Alphanumeric: Alphanumeric data may contain any characters A through Z, 0 through 9, or spaces. Alphanumeric is denoted by the keyword ALPHAN.
Example:

```
27)  GBL-NO;  SIZE = 8; ALPHAN
```

Alphabetic: Alphabetic data may include any combination of letters A through Z, or spaces. The keyword is ALPHAB.
Example:

```
9)  PKG;  SIZE = 2; ALPHAB
```

Numeric: Numeric data can have any combination of characters 0 through 9. A numeric field is a noncomputational field.
Example:

```
26)  REC-SEQ;  SIZE = 3; NUMERIC
```

Real: The data element is identified as a computational number which has an embedded or assumed point. The location of the decimal point must be specified. This data field can include maximum of twelve characters excluding the decimal point but including the sign. The keywords REAL and DEC define real elements.

38) TOT-CHRGE: MAXSIZE = 8; REAL; DEC = 2

In this example total charges are represented by a field length up to and including eight characters with the decimal point located two places to the left of the rightmost digit.

Integer: The data element is assumed to be a computational whole number with no decimal point. Integer elements are defined by the keyword INTEGER. Example:

14) NO-PCS: MAXSIZE = 7; INTEGER

Free Format: When no other data format is specified, any combination of characters will be accepted as a valid data value. This is the default option.

2.2.5 Inversion

Because SHARP uses the partially inverted file concept, any data elements that will be the subject of query searches must be specified by placing the key word INVERT in the data definition statement.

Example:

10) TCN-NO; SIZE = 17; ALPHAN; INVERT

The SHARP DMS includes other types of specifications not presently used in the CONDA data base. These specifications are discussed in the SHARP manual[2], Section 3.

A dictionary of the elements selected for inclusion in the prototype CONDA data base is given in Figure 1.

2.3 INPUT PREPARATION

2.3.1 Modification of Source Data

The CONDA data base comprises several data files, each of which is made up of many logical records. Each record contains the information selected by the user to establish or update the data base. For example, a user may receive a data

Figure 1 - CONDA Data Base Dictionary

ID CONDA

0) SVAN-CNT; MAXSIZE=9; ALPHAN; INVERT	0A (DATA BASE CODE FOR SEAVAN/CONTENTS P0010001)
2) PROC-DATE; SIZE=6; NUMERIC; INVERT	2A SHARP ASSIGNED DATE FOR COMPUTER RUN
3) CONSGR; SIZE=6; ALPHAN; INVERT	3A (CONSIGNOR–SHIPPING ACTIVITY DODAAC)
4) COM; MAXSIZE=3; ALPHAN; INVERT	4A MILSTAMP COMMODITY CODE
5) COM-EXP; SIZE=2; CHAR 1; ALPHAB; CHAR 2; NUMERIC	5A MILSTAMP COMMODITY SPECIAL HANDLING CODE
6) POE; SIZE=3; ALPHAN; INVERT	6A MILSTAMP PORT OF EMBARKATION IDENTIFIER
7) POD; SIZE=3; ALPHAN; INVERT	7A MILSTAMP PORT OF DEBARKATION IDENTIFIER
8) MODE; SIZE=1; ALPHAN	8A IDENTIFIES METHOD OF SHIPMENT USED FOR INITIAL MOVEMENT
9) PKG; SIZE=2; ALPHAB	9A IDENTIFIES TYPE OF PACKING USED FOR CONTAINER/BREAKBULK CARGO
10) TCN-NO; SIZE=17; ALPHAN; INVERT	10A A NUMBER WHICH IS ASSIGNED TO CONTROL A
10) TCN-NO2; SIZE=17; ALPHAN; INVERT	10A SHIPMENT UNIT THROUGHOUT THE TRANSPORTATION CYCLE
11) CONSGE; SIZE=6; ALPHAN; INVERT	11A DODAAC OF THE ULTIMATE RECEIVER OF THE MATERIAL
12) PRI; MAXSIZE=1; INTEGER	12A TRANSPORTATION PRIORITY
13) TAC-CODE; SIZE=4; ALPHAN	13A TRANSPORTATION ACCOUNT CODE–IDENTIFIES SPONSORING SHIPPER
14) NO-PCS; MAXSIZE=7; INTEGER	14A NO. OF PIECES IN THE SEAVAN OR THE SHIPMENT UNIT
15) WEIGHT; MAXSIZE=7; INTEGER	15A TOTAL WEIGHT–IN SEAVAN OR THE SHIPMENT UNIT
16) CUBE; MAXSIZE=7; INTEGER	16A TOTAL CUBE–IN SEAVAN OR THE SHIPMENT UNIT
17) DTE-REC; MAXSIZE=7; INTEGER	17A MATERIAL RECEIPT DATE AT THE POE
18) VSL-STAT; MAXSIZE=2; ALPHAN	18A VESSEL STATUS–IDENTIFIES TYPE OF SHIPPING, TERMS OF CONTRACT
19) DTE-INP; MAXSIZE=4; NUMERIC	19A DATE OF INPUT
20) TYPE-ADDR; SIZE=1; NUMERIC	20A TYPE OF ADDRESS–SHIPPING ADDRESS
42) TYPE-ADD2; SIZE=1; NUMERIC	42A TYPE OF ADDRESS–BILLING ADDRESS
21) ZIPCODE; SIZE=5; NUMERIC; INVERT	21A ZIPCODE OF SHIPPING ADDRESS
43) ZIPCODE2; SIZE=5; NUMERIC; INVERT	43A ZIPCODE OF BILLING ADDRESS
22) AIRTERM-ID; SIZE=3; ALPHAB	22A AIR TERMINAL CODE FOR SHIPPING
44) AIRTERM2ID; SIZE=3; ALPHAB	44A AIR TERMINAL CODE FOR BILLING
24) PORT-CODE; SIZE=3; ALPHAN	24A PORT CODE DESIGNATOR FOR SHIPPING
45) PORT-COD2; SIZE=3; ALPHAN	45A PORT CODE DESIGNATOR FOR BILLING
26) REC-SEQ; SIZE=3; NUMERIC; INVERT	26A RECORD TYPE SEQUENCE–GEN. COMMODITIES–PERSONAL PROP. HOUSEHOLD GOODS
27) GBL-NO; SIZE=8, ALPHAN; INVERT	27A GOVERNMENT BILL OF LADING NUMBER
28) SHIP-PT; MAXSIZE=8; ALPHAB; INVERT	28A IN THE CLEAR SHIPPING POINT–STATE–CITY
28) CONSGR-ADR; MAXSIZE=8; ALPHAB; INVERT	28A SAME AS SHIPPING POINT
29) TYPE MOVE; SIZE=1; ALPHAB; INVERT	29A TYPE MOVEMENT E = EXPORT I OR A BLANK SPACE = IMPORT
30) PICKUP DTE; SIZE=6; NUMERIC; INVERT	30A PICKUP DATE–DATE OF RECEIPT OF SHIPMENT
31) DLVY DATE; SIZE=6; NUMERIC; INVERT	31A DELIVERY DATE OF MATERIAL TO DESTINATION POINT
32) DEST-PT; MAXSIZE=8; ALPHAB; INVERT	32A IN THE CLEAR CONUS DESTINATION POINT–STATE–CITY
32) POE-ADR; MAXSIZE=8; ALPHAB; INVERT	32A SAME AS DESTINATION POINT
33) ORG-DODAAC; SIZE=6; ALPHAN; INVERT	33A ORIGINATING DOD ACT. ADDRESS CODE–RECORD TYPE 301 HOUSEHOLD GOODS
34) DST-DODAAC; SIZE=6; ALPHAN; INVERT	34A DESTINATION DOD ACT. ADDRESS CODE–RECORD TYPE 301 HOUSEHOLD GOODS
35) TOT-WGT, MAXSIZE=7; INTEGER	35A TOTAL WEIGHT–ACTUAL GROSS WEIGHT/NET WEIGHT–APPLIES TO CARGO
36) BILL WGT, MAXSIZE=7; INTEGER	36A ACTUAL BILLING WEIGHT (lb)

Figure 1 (Continued)

```
37) TOT-CUBE; MAXSIZE=5; INTEGER                                    37A ACTUAL TOTAL CUBE (ft3)
38) TOT-CHARGE; MAXSIZE=8; REAL; DEC=2                              38A TOTAL PAID CHARGES (dollars and cents)
39) DEST-GBLOC; SIZE=6; ALPHAN; INVERT                              39A DESTINATION POINT OF ITO GBL RECEIVING ACTIVITY
40) POD-ADR; MAXSIZE=8; ALPHAB; INVERT                              40A IN THE CLEAR OVERSEAS PORT (CITY-COUNTRY)
41) CONSGE-ADR; MAXSIZE=8; ALPHAB; INVERT                           41A IN THE CLEAR OVERSEAS DESTINATION POINT (CITY-COUNTRY)
46) SVAN-CODE; SIZE=1; INVERT                                       46A SEAVAN HEADER RECORD INDICATOR  P = PRIME  O = CONTENTS
38) TOT-CHARGE; MAXSIZE=8; REAL; DEC=2                              38A TOTAL PAID CHARGES (dollars and cents)
50) WGT-TONS; MAXSIZE=6; REAL; DEC=2                                50A (WEIGHT/2000.) SHORT tons
51) MT-CUBE; MAXSIZE=6; REAL; DEC=2                                 51A (CUBE/40.)  MEASUREMENT tons
52) LHRATE; MAXSIZE=5; REAL; DEC=2                                  52A (TOT-CHRGE/TOT-WGT) (dollars and cents/lb)
53) LHCOST; MAXSIZE=8; REAL; DEC=2                                  53A (WEIGHT) * (TOT-CHRGE/TOT-WGT)
60) TYPE-MOV2; SIZE=1; NUMERIC; INVERT                              60A CARGO MOVEMENT INDICATOR  IMPORT = 1, EXPORT = 2
61) CONT-CODE; SIZE=2; NUMERIC; INVERT                              61A COST COMMODITY CODE REFERENCE DA-PAMPHLET 55-3
62) TAC-COD2; SIZE=4; ALPHAN; INVERT                                62A IDENTIFIES SPONSORING SHIPPER SERVICE
63) BILL-MO; SIZE=2; NUMERIC; INVERT                                63A REPORTING MONTH FOR BILLING
64) TRANS-RATE; MAXSIZE=5; REAL; DEC=2; INVERT                      64A TRANSSHIPPED RATE THRU PORT (dollars and cents/lb)
65) TRANS-MTON; MAXSIZE=10; REAL; DEC=3; INVERT                     65A TRANSSHIPPED MEASUREMENT tons THRU THE PORT
66) PREM-RATE; MAXSIZE=5; REAL; DEC=2; INVERT                       66A PREMIUM RATE THRU PORT (dollars and cents/lb)
67) PREM-MTON; MAXSIZE=10; REAL; DEC=3; INVERT                      67A PREMIUM MEASUREMENT tons THRU THE PORT
68) THRU-CHGE; MAXSIZE=10; REAL; DEC=2; INVERT                      68A TOTAL TRANSSHIPPED THRU CHARGES THRU THE PORT
69) PREM-CHGE; MAXSIZE=10; REAL; DEC=2; INVERT                      69A TOTAL PREMIUM CHARGES THRU THE PORT
70) COMMAND; SIZE=1; NUMERIC; INVERT                                70A COMMAND MTMC EASTERN AREA = 1  MTMC WESTERN AREA = 2
71) ORG-MSC; SIZE=4; ALPHAN; INVERT                                 71A MSC ORIGINATING POINT
72) DEST-MSC; SIZE=4; ALPHAN; INVERT                                72A MSC DESTINATION POINT
73) COM-MSC; SIZE=2; NUMERIC; INVERT                                73A MSC COMMODITY CODE
74) WGT-MSC; MAXSIZE=6; INTEGER                                     74A QUANTITY OF CARGO EXPRESSED IN MEASUREMENT TONS
75) COST-MSC; MAXSIZE=7; INTEGER                                    75A MSC COST ROUNDED TO THE NEAREST DOLLAR
76) RDATE-MSC; SIZE=3; NUMERIC; INVERT                              76A REPORTING YEAR AND MONTH (YMM FORMAT)
77) ESTIMATE; SIZE=1; ALPHAN; INVERT                                77A ESTIMATE E = ESTIMATED COST
78) SHIP-DIR; SIZE=1; NUMERIC; INVERT                               78A SHIP DIRECTION IMPORT-EXPORT
79) POE2; SIZE=3; ALPHAN; INVERT                                     6A PORT OF EMBARKATION CODE
80) POD2; SIZE=3; ALPHAN; INVERT                                     7A PORT OF DEBARKATION CODE
81) POE3; SIZE=3; ALPHAN; INVERT                                    81A PORT OF EMBARKATION CODE
82) POD3; SIZE=3; ALPHAN; INVERT                                    82A PORT OF DEBARKATION CODE
83) SEARATE; MAXSIZE=7; INTEGER                                     83A (COST-MSC/WGT-MSC) (dollars/measurement ton)
RELATIONS                                                          999A
COMPUTE LHRATE = TOT-CHRGE / TOT-WGT                               999B
COMPUTE LHCOST = ( WEIGHT ) * ( TOT-CHRGE / TOT-WGT )              999C
COMPUTE WGT-TONS = WEIGHT / 2000.                                  999A
COMPUTE MT-CUBE = CUBE / 40.                                       999B
COMPUTE SEARATE = COST-MSC / WGT-MSC                               999D
$END                                                               999Z
```

tape which has 20 pieces of information for each record but is concerned with only 10 of those pieces for each record. To form the data base the user must input 10 separate lines of data in the proper field positions and two additional lines of data which identify the record key and the transaction date for that record. Each record must be given a unique descriptor, called the record key, which distinguishes it from other records. In addition, each piece of data, called an element, for each individual record must be assigned a number which is unique within the entire CONDA data base. Three of these element numbers, however, are reserved for data necessary to the SHARP system. Element number 0 stores the record key number. Element number 1 stores the overflow of the record key if it exceeds nine characters. Element number 2 stores the date on which the record is introduced into the SHARP system. These elements must be listed at the end of each element key listing for each record key, the 0 element first, element 1 second and then element 2. The listing of elements for that record then follows. The SHARP User's Manual[2] discusses these elements in more detail.

All data may be entered into the CONDA data base in existing form with two exceptions.

(1) There may be no embedded blanks within the data. For example, if a number was in integer format on the data tape in an allotted field size (on the tape) of 10 columns, but the actual number had only three digits, it would appear as bbbbbbb123. The data read into the data base in this format would be unacceptable.

(2) If a data field from the input tape contains a negative value, the complement of the negative number must be input. A simple algorithm is used to find the complement of that negative data value:

$$\text{Correct data value format} = 10^{(\text{field size} - 1)} + (\text{data value})$$

The CONDA system will be able to use this value for a negative number and subsequently change the data value back to a negative number on output.

The CONDA system also includes a format checking program. As data are entered in the standard transaction format, they are checked for validity in accordance with the input specifications. This capability eliminates the need for deleting bad data and re-entering correct data.

After the analyst has defined the new set of unique elements for his new data, the CONDA system will read the new data, validate the format, and through the use of the loader setup, add the new data to the existing data base. All data elements found to be incorrect will be rejected. A sample data transaction including coded cards and a few data elements, is presented in Table 1. The first line (beginning with A) represents an "add" transaction for record number 121 followed by the data for data elements 4, 5, and 6. The third line (beginning with the C) represents a "change" transaction for record number 121. The value in element 5 was changed to MC from MV.

2.3.2 Input Formats

To prepare data and load it into the SHARP Data Management System requires the use of a formatting program. Figure 2 shows the input card configurations. The first input card, the program identification and date card, identifies the data base name and the date on which data will be added to the data base. The second input card, INVRT, fills the array used to specify whether or not a data element is to be inverted. The inversion capability must be specified when data files are loaded into the CONDA data base master file. A blank in the specified field indicates that the data element is to be inverted; an X in the specified field indicates no inversion. The CONDA system then uses a direct access technique to retrieve the records in a file which satisfy a user query on the file.

The MSIZE card describes the field length for each data element. The fourth input card, defining the ICHAR array, reads the actual input data from the data tape for each data element defined. The program listing of Figure 3 indicates that the card that reads the values for the ICHAR array is in a major loop. All input data values are read, checked for the proper format (no blanks or negative numbers), and written to the output files. This output file will later be loaded into the SHARP Data Management System.

The ICNT array, Figure 3, defines the data element numbers that the user will input into the data base. For the example shown as part of Figure 3, the data element numbers are 76, 81, 82, 71, 72, 73, 74, 75, 77, and 78.

TABLE 1 - SAMPLE DATA TRANSACTION

```
ATTACH,SISFILE,CONDADATADEFFILE,ID=SHRP,MR=1.

ATTACH,UPDLIB,ID=SHRP,MR=1.
LIBRARY,UPDLIGB.
FMCK.
7/8/9
ID CONDA
A      121    4)  AC     5)  MV     6)  340    $END
A       50    4)  AR     5)  MP     6)  279    $END
C      121    5)  MC     $END
$STOP
```

1. | ID | MDATE | |
|---|---|---|
| | 11 | 21 |

Variable	Format	Description
ID	A5	Program ID
MDATE	A6	Date

2. | INVRT(I),I=1,10 | |
|---|---|
| | 11 |

Variable	Format	Description
INVRT	10A1	An "X" in field means element NOT to be inverted

3. | MSIZE(I),I=1,10 | |
|---|---|
| | 11 |

Variable	Format	Description
MSIZE	10I2	Describes Field Length for each element

4. Input Data Card (for Cost Accrual File)

ICHAR(I),I=1,10

Variable	Format	Description
ICHAR (1)	A3	Element Number 76
(2)	A3	81
(3)	A3	82
(4)	A4	71
(5)	A4	72
(6)	A2	73
(7)	I7	74
(8)	I8	75
(9)	A1	77
(10)	A1	78

Figure 2 - Description of Inputs

Figure 3 - Data Tape Format Routine

```
      PROGRAM DBASF5(INPUT=128,OUTPUT=128,TAPE5=INPUT,TAPE6=OUTPUT,
     * TAPE1=/144,TAPE2)
      DIMENSION ICHAR(10),INVPT(10),MSIZE(10),ICNT(10)
     DATA ICODE/1H3/,IBLK/1H /
      DATA ICNT/4H0C76,4H0081,4H0082,4H0071,4H0072,4H0073,4H0074
     * 4H0075,4H0077,4H0078 /
       NREC=0
      IDUM = 000009000
      READ(5,1000) ID, MDATE
      READ(5,1001) (INVRT(T),T=1,10)
      READ(5,1002) (MSIZE(I),(I = 1,10)
     WRITE(2,2000) ID
   10 CONTINUE
      READ(1,1003) (ICHAR(I),(I=1,5) (ICHAR (I),I=6,10)
      IDUM = IDUM + 1
     NREC=NREC+1
      IF(EOF(1).NE.0.0) GO TO 999
      DO 100 I = 1,10
      IF(I .EQ. 7 .OR. I .EQ. 8) GO TO 40
     IF(IBLK.EQ.ICHAR(I) GO TO 100
      MASK=770000000000000000008
      ITEMP = ICHAR(I)
      DO 20 J = 1,10
      ITEST = (MASK .AND. TTEMP)
      IF(ITEST .NE. 1L .AND. ITEST .NE. 1L0) GO TO 30
      ITEMP = SHIFT(ITEMP,6)
   20 CONTINUE
C    ALL BLANK OR ZERO
      GO TO 100
   30 CONTINUE
C    NOT BLANK OR ZERO
      WRITE(2,2006) IDUM,ICODE,ICNT(I),MSIZE(I),ICODE,ICNT(I).
     * INVRT(I),ICHAR(T)
```

Figure 3 (Continued)

```
      GO TO 100
   40 IF(ICHAR(I) .EQ. 0) GO TO 100
      IF(ICHAR(I) .GT. 0) GO TO 80
C     DATA IS A NEGATIVE NUMBER
      IDID = (10**(MSIZE(I) - 1)) + ICHAR(I)
      IF(I .NE. 7) GO TO 45
C     DATA .LE. 0 I = 7
      WRITE(2,2004) IDUM,ICODE,ICNT(I),MSIZE(I),ICODE,ICNT(I).
     * INVRT(I),IDID
      GO TO 100
   45 CONTINUE
C     DATA .LE. 0 I = 8
      WRITE(2,2009) IDUM,ICODE,ICNT(I),MSIZE(I),ICODE,ICNT(I).
     * INVRT(I),IDID
      GO TO 100
   80 CONTINUE
C     DATA .GT. 0 --- NOW IS THIS 7 OR 8
      IF(I .NE. 7) GO TO 82
C     DATA .GT. 0  I = 7
      WRITE(2,2007) IDUM,ICODE,ICNT(T),MSIZE(I),ICODE,ICNT(I).
     * INVRT(I),ICHAR(I)
      GO TO 100
   82 CONTINUE
C     DATA .GT. 0 I = 8
      WRITE(2,2008) IDUM,ICODE,ICNT(I),MSIZE(I),ICODE,ICNT(I).
     * INVRT(I),ICHAR(I)
  100 CONTINUE
      WRITE(2,2002) IDUM,ICODE,ICODE,IDUM
      WRITE(2,2003) IDUM,ICODE,ICODE,MDATE
      GO TO 10
  999 WRITE(2,2005)
      ENDFILE 2
```

Figure 3 (Continued)

```
      WRITE(6,3000) NREC
      STOP
1000  FORMAT(A5,5X,A6)
1001   FORMAT(10A1)
1002   FORMAT(10I2)
1003   FORMAT(A3,24X,2A3,4X,2A4,3X,A2,1X,I7,3X,I8,5X,A1,6X,A1)
2000  FORMAT(21X,1H*,9X,A5)
2001  FORMAT(I9 ,3X,A1,A4,2X,A2,A1,A4,4X,A1,A10)
2002   FORMAT,I9,3X,A1,*0000  09*,A1,*0000*,5X,I9,9)
2003   FORMAT,(I9,3X,A1,*0002  06*,A1,*0002*,4X,*X*,A6)
2004   FORMAT(I9,3X,A1,A4,2X,I2.2,A1,A4,4X,A1,*-*,I6)
2005  FORMAT(4H$END )
2006   FORMAT(I9,3X,A1,A4,2X,T2.2,A1,A4,4X,A1,A10)
2007   FORMAT(I9,3X,A1,A4,2X,T2.2,A1,A4,4X,A1,I7.7)
2008   FORMAT(I9,3X,A1,A4,2X,T2.2,A1,A4,4X,A1,I8.8)
2009   FORMAT(I9,3X,A1,A4,2X,I2.2,A1,A4,4X,A1,*-*,I7)
3000   FORMAT(*  RECORDS = *,I7)
      END
```

2.4 LOADING THE DATA BASE

When the data have been properly formatted, the user is ready to load the data into the CONDA Data Base. The formatted data resides on a disk file to be attached by the loader control cards. The Loader setup is the same for all types of data to be input. Figure 4 illustrates the required input configuration for loading the data base.

2.5 EXAMPLE OF LOADING AND MODIFYING THE DATA BASE

Figure 5 is an example of the data setup. A Navy installation provides a data tape with X number of records stored on it. Each record contains 20 pieces of data, of which ten are to be analyzed. Ten unique element numbers are assigned to these ten pieces of data.

In Figure 5, the first nine characters of each line represent the record key. The record key remains the same for all data items of a given record. The first five lines contain data for the same record, whose record key is 9001. The next three spaces are blank.

The 3 in column 13 represents an "add" transaction which inserts a new logical record into the existing data base. The "add" transaction is one of three types of transactions used by the SHARP System. The other two transactions are "change" and "delete". Section 4 of the SHARP Manual[2] gives further explanation of these three transactions. This example indicates all four records are being added to the SHARP Data Base.

The next four columns (Cols. 14 through 17) contain the data element number of the piece of data that follows. Two more blank spaces (Cols. 18 and 19) are followed by a field two columns long (Cols. 20 and 21) which contain the length of the data for this element. Figure 5 indicates that the lengths of the first three pieces of data for record key 9001 are 3, 8, and 1 columns.

The next five columns (Cols. 22 through 26) repeat columns 13 through 17. The next four columns (Cols. 27 through 30) are blank. The next one-column field (Col. 31) contains either a space or an X to specify whether this data element is to be inverted. Figure 5 shows that only two pieces of data in record 9001, elements 75 and 2, are not be inverted.

The last field, starting in column 32, contains the data. The field length may be 1 - 17 columns long and may be alphanumeric.

Figure 4 - SHARP Loader Card Setup

```
PAUSE, JOB REQUIRES DISK DV4794. (your disk pack)
MOUNT,VSN=DV4794,SN=CONDAPK.
MAP(OFF)
LIMIT,13000.
ATTACH,INDISC,CABQCOSTO,ID=CABQ.                    Attach data file
FILE,INDISC,RT=Z,BT=C,MRL=127,MBL=640.
FILE,OUTDISC,RT=F,BT=C,FL=127,MBL=640               Change source data format to a format
COPYRM,INDISC,OUTDISC.                              acceptable to SHARP
REWIND,OUTDISC.                                     Rewind data file
ATTACH,UPDLIB,ID=SHRP,MR=1.
LIBRARY(UPDLIB)                                     Attach and define SHARP Library routines
REQUEST,SH00101.*PF.                                Request permanent file
HSORT.                                              Sort data by transact or codes and date
CATALOG,SH00101,CDASH00101,ID=CAFA.                 Create permanent file of sorted data
REQUEST,SH00201,*PF,SN=CONDAPK.
REQUEST,COPY109,*PF.                                Request permanent files
ATTACH(DIRIN,CDADIR,ID=SHRP,SN=CONDAPK)             Attach Direct Master File
RENAME(DIRIN,,CY=2)                                 Rename above File to cycle '2' (from cycle 1)
COPYBF(DIRIN,SH00201)                               Copy Master File to previously requested PF
REWIND(SH00201)                                     Rewind new file
DUPD.                                               Execute Update program for Direct Master PF
REWIND(ZZZZZEF)                                     Rewind error file
CATALOG,SH00201,CDADIR,ID=SHRP,CY=1.
```

Figure 4 (Continued)

```
RETURN,SH00201.                          Create new Master File and return to system
COPYCF(IN00109,COPY109)                  Copy Inverted Master file and rewind
REWIND(IN00109)
CATALOG,COPY109,CDCACOPY109,ID=CAFA.     Create new Inverted Master File
ATTACH,INSM,ID=SHRP,MR=1.                Attach sort Parameters
REWIND(IN00109,INVTRAN)                  Rewind these files
FILE(IN00109,BT=C,RT=F,FL=88,MBL=640)
FILE(INVTRAN,BT=C,RT=F,FL=88,MBL=640)    Format these files
LOSET(FILES=IN00109/INVTRAN)             Load these files into memory
SORTMRG(I=INSM)                          Sorts the transactions out of direct update
REQUEST,IN00101,*PF,SN=CONDAPK.          Request permanent file on personal discpack
ATTACH,INVIN,CDAINV,ID=SHRP,SN=CONDAPK.  Attach Inverted Master File
RENANE(INVIN,,CY=2)                      Rename above file to cycle '2'
COPYBF(INVIN,IN00101)                    Copy this file
REWIND(IN00101)                          Rewind this file
IN1.                                     Execute Update program for Inverted M.F.
CATALOG,IN00101,CDAINV,ID=SHRP,CY=1.     Create next Inverted Master File
PURGE(SH00101)
PURGE(COPY109)
PURGE(DIRIN)
PURGE(INVIN)                             Purge these four files
EXIT(S)                                  In case of error in previous execution, pass
REWIND(ZZZZZEF)                          Control to this point-otherwise skip
```

Figure 4 (Continued)

```
COPYBF(ZZZZZEF,OUTPUT                                  Rewind error file and copy to output
    PUT YOUR UPDATES HERE
STOP                                                   Stop
ATTACH,SISDISC,DATADEFFILE,ID=SHRP,MR=1.               Attach data definition file for
                                                       validating data
ATTACH,FMCKL,PHILFMCK,ID=SNAI,MR=1.                    Attach Validation program
FILE(INPUT,LFN=INDISC)                                 Define input file for Validation program
LDSET(FILES=INPUT)                                     Load input file into core
FMCKL.                                                 Execute Validation program
RETURN,SISDISC.                                        Return data definition file to system
```

Figure 5 - Data Input Example

Card Columns

1 9	13 17	20 22 26	31 32 48
		*	
000009001	30076	0330076	706
000009001	30075	0830075	X00003684
000009001	30078	0130078	1
000009001	30000	0930000	000009001
000009001	30002	0630002	X780920
000009002	30076	0330076	706
000009002	30081	0330081	3DS
000009002	30082	0330082	BA1
000009002	30073	0230073	60
000009002	30074	0730074	X0000098
000009002	30075	0830075	X00004702
000009002	30078	0130078	1
000009002	30000	0930000	000009002
000009002	30002	0630002	X780920
000009003	30076	0330076	706
000009003	30081	0330081	3DS
000009003	30082	0330082	BA1
000009003	30071	0430071	AD17
000009003	30072	0430072	EQ01
000009003	30073	0230073	60
000009003	30074	0730074	X0000098
000009003	30075	0830075	X00000110
000009003	30078	0130078	1
000009003	30000	0930000	000009003
000009003	30002	0630002	X780920
000009004	30076	0330076	706
000009004	30081	0330081	CK1
000009004	30082	0330082	1R5

Figure 5 (Continued)

```
000009004       30073       0230073        84
000009004       30074       0730074        X-999990
000009004       30075       0830075        X-9999862

000009004       30077       0130077        E
000009004       30078       0130078        2
000009004       30000       0930000        000009004
000009004       30002       0630002        X780920
```

After all ten elements have been listed, two more lines of data must be added for elements number 0 and 2 which contain the record key and the date this record key was added to the CONDA Data Base, respectively. In the first record of the example, 9001, element number "0" contains the record key itself and element number "2" the date, "September 20, 1978", this record was added to the CONDA Data Base. The setup for these last two lines of each record is similar to the rest of each record.

The first nine columns of both the element 0 and 2 contain the record key followed by three columns of spaces. The following column contains a "3" followed by "0000bb09" for element number "0" - the record key and "0002bb06" for element number "2" - the date this record key was created. In the following five columns (Cols. 22 through 26) repeat columns 13 through 17 for both elements. For element "0" skip the following five columns (Cols. 27 through 31) and then list the record key itself. For element "2" skip the following four columns (Cols. 27 through 30) and then place an "X" in column 31 followed by the current date. This date must be listed by year, month, and day allowing two columns for each (six total) occupying columns 32 through 37. This process must be repeated for each record key to be added to the SHARP Data Management System.

SECTION 3
QUERYING THE CONDA DATA BASE

3.1 CAPABILITIES

3.1.1 Overview

Users may interrogate the CONDA Data base in either batch or interactive mode using an English-like command language. Questions are constructed which specify the desired selection criteria, and those records in the data base which meet the selection criteria wil be retrieved and output. Either a print statement or a previously defined report statement may be used to specify how the output information will appear. Section 4 describes seven previously defined CONDA output reports.

For each query, the system will determine the number of data base records (hits) that satisfy the specified selection criteria. The user then has the option of specifying for each question whether the selected records will be printed on line on the user terminal, off line at a batch printer, or not printed at all.

A maximum of 75 questions may be entered at one time in either the interactive or batch mode. With each question a different sort, print, or report specification may be entered. Each question is analyzed for errors. The query values are checked to ensure that the format agrees with that of the data base definition for the specified data elements. Questions are also checked for query language syntax errors, and appropriate diagnostics are output if errors are found. During an interactive query, error diagnostics for each question are output immediately to the user. The user may then re-enter the information correctly before entering the next question.

Because the data base is too large to be maintained on line, and because most query results require a large printer output, CONDA is designed to run both off line and via on-line batch terminal.

3.1.2 Batch Terminal Operation

The batch terminal mode of operation, the user enters the query at the user terminal and logs off. The computer processes the query while the user is off-line. The query is entered into a batch queue where it is executed in batch mode along with other batch jobs according to priority. The user can retrieve the

output search report in approximately 30 minutes to 1 hour depending on the backlog of jobs in the computer. (There is a significant cost advantage to using this retrieval method as opposed to the interactive retrieval method.) The user has the option of retrieving his output either from the off-line printer at the computer center or at the user terminal. If the user specifies that the output is to be printed at the computer center, the system requires no further action by the user after he logs off. If the user desires his output printed at the user terminal, he logs in again approximately one half hour later using the QUERY option. The system automatically checks its files to determine whether the user has an active request outstanding. If the active request has not been processed completely at the time the user logs in, the user logs in again at a later time. A new request cannot be initiated until the active request is completed. If the active request has been processed and completed at the time the user logs in, the user is given the number of hits for each question in his query. He then designates whether he wants to delete the question, print the output at the teletype, or print the output off-line at the computer center. The user then logs out if he has no additional queries. The query procedures is as follows:

1. Log On
2. Select Data Base File
3. Enter Query
4. Specify on-line or off-line output
5. Log Off
6. (After Job Execution)
   If off-line output was specified, receive output from off-line printer.
7. If on-line output was specified, log back on approximately one-half hour after the initial query was made.
8. Receive number of hits for each question. Specify output disposition for each question.
9. If on-line output was specified, receive output at terminal.
10. Log Off
11. If off-line output was specified, receive output from printer.

The CONDA Program user can access the computerized Data Base directly if he has a User ID and a PASSWORD. Each user is assigned a unique user ID by the DTNSRDC User Services Group. The user's PASSWORD is an active DTNSRDC job number to which computer costs are charged. Here is an example of the user access procedure.

```
NSRDC 66ØØ INTERCON U 4.6
DATE  Ø4/Ø3/79
TIME  1Ø.Ø9.58.
LOGIN,CAFAMYERS,SUP
XXXXXXXXXX ENTER ACCESS NUMBER-
COMMAND- COMRADE,SHARP

COMRADE       TIME:      1Ø.11.49.
              DATE:      Ø4/Ø3/79

SHARP SUBSYSTEM ENTERED
????QUERY
```

3.1.3 Off-Line Operation

Users may also wish to submit queries off-line via a punched card deck. In this case, the query questions can be punched on cards using the same command language as for on-line queries.

3.2 QUERY CONSTRUCTION

3.2.1 Search Acronyms

The data names assigned in the CONDA Data Definition are used in CONDA queries. These acronyms are used in referencing data elements in a query and are called "search acronyms."

3.2.2 Search Criteria

In any query the search criteria qualify the search. Table 2 gives a complete list of available search criteria. Each criterion has one or more of allowable

TABLE 2 - CONDA SEARCH CRITERIA

CRITERIA	* ENTER CODE
Logical Operators	
**Is; Equal to	IS IS EQ TO IS EQUAL TO IS EQUALS TO IS = TO
Not Equal To	IS NOT IS NOT EQUAL TO IS NE TO
Less Than (but not equal to)	IS LT THAN IS LS THAN IS LESS THAN IS < THAN
Less Than or equal to	IS LE TO IS LESS OR EQUAL TO IS LESS THAN OR EQUAL TO
Greater Than (but not equal to)	IS GR THAN IS GT THAN IS GREATER THAN IS > THAN
Greater Than or Equal to	IS GE TO IS GREATER OR EQUAL TO IS GREATER THAN OR EQUAL TO
Ranging Operators	
Between but not equal to	IS BN TO IS BETWEEN BUT NOT EQUAL TO IS BETWEEN
Between or Equal to	IS BE TO IS BETWEEN OR EQUAL TO
Between or Equal to higher value	IS BH IS BETWEEN OR EQUAL HIGHER
Between or Equal to lower value	IS BL IS BETWEEN OR EQUAL LOWER

TABLE 2 (Continued)

Special Operators	
Prefix	PREFIX
Suffix	SUFFIX
Present	PRESENT
Absent	ABSENT

* Underlined Codes are optional

** Default Search Criteria, used when no criterion is specified

syntax combinations for specifying the criterion in a query. The underlined portion of the codes are optional and are allowed for readability.

3.2.3 Search Values

The search value is that value to which the search acronym is compared in the search. For example, in the MTMC Cargo File of the CONDA Data Base, the data element POE is compared on equality (EQ) to the search value lMl. This means that in a designated search all values of POE equal to lMl would be selected.

3.2.4 Query Format and Restrictions

A CONDA query is composed of one or more questions. Each query is introduced by the Syntax - QUERY XXX where XXX is a two or three character user identification tag such as personal name initials. Each question is introduced by the conjunction IF and is terminated by either a semicolon or a question mark. Each question may contain one or more independent clauses which contain as a minimum a search acronym, a search criterion, and one or more search values. An independent clause is introduced by one of the conjunctions IF, AND, OR, OR IF. Each sentence within a question except the last may be optionally terminated with either a semi-colon or question mark. The last sentence in a question must be terminated by either a semi-colon or question mark. The command $END must be entered following the last question in a query.

Example:

```
QUERY FAM
IF POE IS lP2;

IF POE EQ lP2 OR lMl?

IF POE IS lP2
OR POD IS CEl;

IF POE IS lMl OR lP2
AND POD IS CEl;
$END
```

This query is composed of four questions. The first question consists of an acronym (POE), a search criterion (IS), and one search value (1P2). The second question has one sentence but two search values (1P2 and 1M1). The third question contains two sentences, each of which has one search value (1P2 in sentence one, "CE1" in sentence two). The two sentences are joined by the conjunction OR. Question four contains two sentences with a total of three search values, two in sentence one, and one in sentence two. Note that the rule of conjunction OR is different in questions three and four. In question three, OR introduces a sentence whereas in question four OR joins two search values.

The query language syntax closely approximates the way questions would be phrased in English. In question one, all POEs which have a value of 1P2 (Charleston, SC) would be selected. In question two, the ports of embarkation 1P2 and 1M1 (Charleston, SC, and Norfolk, VA,) would be selected. In question three, the data base records selected are those whose POE is 1P2 and/or whose POD is CE1. In question four, the records selected would be only the POEs of Norfolk or Charleston whose POD CEI (Guantanamo Bay). Codes for all ports are provided in DoD Regulation 4500.32-R, Volumes I and II.

Compound search values can be expressed in many acceptable ways in the query language. Section 5.1.4.1 of the SHARP Manual[2] provides examples of format variations.

Queries to the CONDA Data Base are entered in a free-form format. Multiple spaces are allowed between words; however, at least one space is required. Spacing on either side of punctuation marks is optional. Any number of words may be entered on a line; however, on batched type queries using punched card input only the first 72 characters are used. A maximum of 15 sentences and 25 search values may be entered per question. Search values may not exceed 50 characters in length including embedded blanks. Users are restricted to a maximum of 25 questions input at one time in either the interactive or batch mode.
Examples:

1. Correct
```
QUERY FM
IF COM IS 102;
```
2. Incorrect
```
IFCOM IS 102;
```
3. Correct
```
IF TYPE-MOV2 IS 1;
```
4. Incorrect
```
IF TYPE - MOV2 IS 1;
```

5. Correct IF POE IS 1K2 OR 1MJ;

6. Incorrect IF POE IS 1K2 OR1MJ;

Question 2 is incorrect because at least one space was not entered between IF and COM. Question 4 is incorrect because embedded blanks are not allowed in acronyms. Question 6 is incorrect because at least one space was not entered between the conjunction OR and the search value 1MJ.

3.2.5 AND/OR Dominance

If both conjunctions (AND,OR) appear in a question separating the conditions of a question, the conjunction AND is dominant over the OR.

Example:

```
IF REC-SEQ IS 101
AND SHIP-PT IS VANORFOL
OR DEST-PT IS SCCHARLE;
```

In this case, if OR was dominant, records would be selected where DEST-PT is SCCHARLE regardless of whether any other conditions are met. However, since AND is dominant, the logic is grouped as follows:

```
IF REC-SEQ IS 101
AND     SHIP-PT IS VANORFOL
        OR DEST-PT IS SCCHARLE;
```

For a record to be selected, REC-SEQ must be 101 and in addition either of two conditions must be met -- SHIP-PT must be VANORFOL or DEST-PT must be SCCHARLE.

When both AND and OR are used as conjunctions introducing sentences, the OR condition can be made dominant by using OR IF.

Example:

```
IF REC-SEQ IS 101
AND SHIP-PT IS VANORFOL
OR IF DEST-PT IS SCCHARLE;
```

In this example, the logic grouping is as follows:

```
IF      REC-SEQ IS 101
        AND SHIP-PT IS VANORFOL
OR IF   DEST-PT IS SCCHARLE;
```

The use of the OR IF changes the logic such that a record is selected if DEST-PT is SCCHARLE regardless if any other conditions are met.

3.2.6 Ranging Searches

The ranging search capability allows data value ranges to be specified rather than single data values. The ranging search criteria are listed as part of Table 2.

Examples:

1) IF COM IS BE 100/125?

In this case all records with a commodity code (COM) in the range of 100 to 125 inclusive will be selected.

2) IF COM IS BN 100/125?

In this example commodities 101 to 124 inclusive will be selected.

3) IF COM IS BH 100/125?

This example specifies all commodities in the range 101 to 125 inclusive.

4) IF COM IS BL 100/125?

This final example selects all commodities in the range 100 to 124 inclusive.

3.2.7 Special Operators

Queries can be made on the beginning (left most) characters, the middle characters, or the end characters of a data element. These three operators are defined as a PREFIX, PARTIAL, or SUFFIX search, respectively.

For example, assume that a date is entered into a file in the numeric form YYMMDD (YY-year, MM-month, DD-day). The acronym for this date is given as DATE. If the user wants to select all records for the year 1978, his query would be

IF PREFIX DATE IS 78;

If the user wants to select records from June, his query structure would be

IF PARTIAL DATE IS (2)06;

This query is interpreted as follows: skip the first two characters of DATE and compare the next two characters on the value 06. All records with dates in the month of June will be selected.

Finally, if the user wants to select all records which reflect transactions on the first day of the month;

IF SUFFIX DATE IS 01;

This query satisfy the first day of the month requirement.

Two other special operators include a present and an absent search. The present search locates all records in the data base which have any values entered for the given data element.

Example:

```
IF COM IS PRESENT;
```

The absent search is the opposite of the present search.

Example:

```
IF COM IS ABSENT;
```

3.3 SORT SPECIFICATION

The CONDA system can, on request, sort output records selected during a search. Any combination of data elements defined in the CONDA Data Base may be specified for the sort sequence. The data elements may be sorted in both ascending (low to high values) and descending (high to low) sequence. Ascending order (low to high) is the default sequence.

Examples:

```
QUERY FAM
IF COM IS BE 360/399;
SORT ON COM.

IF POE IS IMJ
AND POD IS PRESENT;
SORT ON POD.
SEND
```

The results of the first question would be sorted in ascending order beginning at 360 and ending with 399. The second question results in a sort of the PODs in alphabetical order.

The prefix and partial sort capabilities allow the Data Base user to sort on the left-most characters or characters in the middle of a data element.

Example:

```
1)  SORT ON (2) DATE.
```

In this example; the sort would be on the first two characters of DATE. For the purpose of this example, assume DATE has a format YYMMDD.

A partial sort allows the user to sort on the middle characters of a data element.

Example:

```
1)  SORT ON (*16,1) TCN-NO.
```

3.4 PRINT SPECIFICATION

With each question the user may enter a print statement which specifies the data elements to be output from each record selected in the search. A print statement may be entered either in a standard system print format or in a report statement. Examples of the standard system print format are:

```
QUERY FAM
IF POE IS IMJ AND POD IS CE1;
PRINT NO-PCS, WEIGHT, AND CUBE.

IF COM IS BE 360/399;
PRINT COM, AND NO-PCS
$END
```

The standard system print format is described in more detail in Section 5 of the SHARP Manual[2]. The report definition capability is discussed in detail in Section 4 of this report.

3.5 COMPUTATION CAPABILITY

Computations may be performed on data elements in those records selected in the search. Table 3 gives a list of computation specifications. The following examples illustrate computation specifications:

1)
```
IF REC-SEQ IS 301 AND SHIP-PT IS SCCHARLE;
PRINT BILL-WGT.
SUM BILL-WGT.
```

2)
```
IF REC-SEQ IS 301 AND SHIP-PT IS SCCHARLE;
PRINT AVERAGE TOT-CUBE.
```

3)
```
IF POE IS 1P2 AND POD IS PRESENT.
SORT ON POD.
SUBTOTAL NO-PCS, WEIGHT, AND CUBE ON POD.
```

In Question 1, BILL-WGT is to be output and summed. Any number of data elements may be entered and summed following the keyword SUM.

Question 2 will print the average of TOT-CUBE for household goods (REC-SEQ IS 301) are shipped from Charleston, South Carolina.

TABLE 3 - COMPUTATION SPECIFICATIONS

Computation	Acronym(s)
Sum or total a data element	SUM TOTAL
Average of a data element	AVG AVERAGE
Percent (1 element - another x 100)	PERCENT
Percent sum (sum two elements then compute percentage)	PERCENT SUM TOTAL
Maximum of a data element	MAXIMUM MAX
Minimum of a data element	MINIMUM MIN
Standard deviation of a data element	SIGMA
Count the occurrences of a data element	COUNT
Sub-total a data element when another element changes in value	SUBTOTAL SUBTOTALS SUB-TOTAL

Question 3 allows the NO-PCS, WEIGHT, and CUBE to be subtotaled each time the POD changes. The net result of this query is a total of NO-PCS, WEIGHT, and CUBE leaving from port 1P2 and going to all possible PODS.

The other computation specifications operate in the same way.

3.6 REPORT SPECIFICATION

The report specification enables the user to specify that his output be printed in a previously defined report format. If no report statement is entered, the output will be displayed in the standard system print format. The following examples illustrate the report specification:

```
QUERY FM1.
IF POD IS CE1;
SORT ON POE.
REPORT RØ1.
IF SHIP-PT IS INCRANE AND
DEST-PT IS NJEARLE;
REPORT R04.
$END
```

The records selected for output from the first question will be displayed in a report format which was catalogued under the name R01. A report statement is always introduced with the keyword REPORT. The report identification, in this case R01, immediately follows. A different report may be specified for each question. The second example will display records selected for output in a report format which was catalogued under the name R04. Seven previously defined reports are described in Section 4.

SECTION 4
CONDA REPORTS

4.1 CAPABILITIES

The SHARP Data Management System has on-line and off-line report definition capability. In addition, any number of reports may be defined at any one time. If the user makes errors during the defining of a report, error messages will be output. The user may then re-enter the report definitions correctly.

Each report definition is catalogued under a specified identification acronym. The user selects the report at query time by specifying the report ID. A user may also define a report and select it in a query during the same terminal session. The report IDs which are already defined for the CONDA Data Base are REPORT R01 through R07 which are defined in Section 4.3.

Both columnar- and row-type reports may be defined. The information from one data base record appears on a single line on the output page of a columnar report. All seven previously defined CONDA reports are of the columnar type. Row reports differ from columnar reports in that multiple lines are required on the output report page to contain the information from one record.

The initial input for all the CONDA report definitions is the field ID followed by the data base file for which the report is defined (e.g., ID CONDA). The file ID is followed by the report ID in the form REPORT XXX, where XXX is a one to three-character identification code assigned by the user. This sequence is followed by specific report definition information.

4.2 REPORT DEFINITION PROCEDURE

Columnar reports are normally defined when the information to be selected from a data base record can be contained on a single line of the output page. Column reports are defined using the keywords COL HEAD or COLHEAD which introduce column headings for the report. The following input illustrates the definition of a typical CONDA report:

```
ID CONDA
REPORT RØ2
PAGE WIDTH = 135
MHEAD = $SEAVAN COST SUMMARY$
COL HEAD 1 = $/CL CONSGR$
```

```
COL HEAD 2 = $/CL POE$
COL HEAD 3 = $/CL POD$
COL HEAD 4 = $TOT WGT/(TONS)$
COL HEAD 5 = $/TOT CUBE$
COL HEAD 6 = $/MTONS$
COL HEAD 7 = $/TOT-CHARGE$
COL HEAD 8 = $LH RATE/PER LB$
PRINT SHIP-PT, POE-ADR, POD-ADR, WGT-TONS,
CUBE, MT-CUBE, TOT-CHARGE, LHRATE
$END
```

The ID indicates the CONDA data file. Report R02 has been assigned as the identification code for this report. PAGE WIDTH indicates the number of characters or spaces to an output line. A page width of 72 characters is normally used for remote terminal reports. The report master heading, MHEAD, provides a title for the report. An option allows three variable headings following the master heading to annotate the output report. The keyword HEAD introduces each of those variable headings which may have from one to sixty characters and must be bounded by either a single quotation mark or a dollar sign. The heading number immediately follows the keyword HEAD or HEADING.

Following the master heading (and subheads if used) are the column headings for the respective data columns. The left-most column heading for the report page is entered first (COL HEAD 1). Each column heading is bounded by either single quotation marks or single dollar signs. A maximum of three column heading lines per column heading may be entered with a maximum of twelve characters per column heading line. Slashes are used to separate column heading lines. The first column of the example used consists of one heading line. The use of a slash drops the heading one so that it is printed on the second line used for headings.

Following the eighth column heading in the example, a PRINT statement lists the data elements to be output under the column headings. Each element is entered via its acronym, the first element to be output under column heading one, the second under column heading two, etc.. The report definition is terminated by $END.

If a column report is defined which is too wide for the defined page width, an error message will be returned. The following procedure should be used to compute the minimum report page width:

1) For each data column, take the number of characters in the widest column heading line or the maximum number of characters which may be output for a data value, whichever is larger.

2) Sum the numbers computed for each data column.

3) Add one space for each column except the last to determine the minimum spacing between columns.

4) Compare the total spaces required to the page width. If the total spaces are greater than the page width, the report definition is invalid and must be redefined.

4.3 PREDEFINED CONDA REPORTS

Seven reports (see Figures 6-12) have been predefined for the present prototype CONDA data base. These formats represent the kinds of reports that can be generated from the data base.

```
ID CONDA
REPORT R01
PAGE WIDTH = 135
MHEAD = $NAVY CARGO MOVEMENT LISTING$
COL HEAD  1 = $/CONSGE$
COL HEAD  2 = $/POE$
COL HEAD  3 = $/POD$
COL HEAD  4 = $/CONSGE$
COL HEAD  5 = $/COM$
COL HEAD  6 = $COM/EXP$
COL HEAD  7 = $/MODE$
COL HEAD  8 = $/PKG$
COL HEAD  9 = $/TCN NUMBER$
COL HEAD 10 = $/PRI$
COL HEAD 11 = $TAC/CODE$
COL HEAD 12 = $NO OF / PIECES$
COL HEAD 13 = $/WEIGHT$
COL HEAD 14 = $/CUBE$
COL HEAD 15 = $REC/DATE$
COL HEAD 16 = $VSL/STATUS$
PRINT CONSGR, POE, POD, CONSGE, COM, COM-EXP, MODE,
PKG, TCN-NO, PRI, TAC-CODE, NO-PCS, WEIGHT, CUBE,
DTE-REC, VSL-STAT
$END
```

The data elements in this report are: consignor, port of embarkation, port of debarkation, consignee, commodity, commodity exception, mode, package code, transportation control number, priority, transportation account code, number of pieces, weight, cube, date received at POE, and vessel status code.

Figure 6 - Navy Cargo Movement Report

```
ID CONDA
REPORT RO2
PAGE WIDTH = 135
MHEAD = $SEAVAN COST SUMMARY$
COL HEAD 1 = $/CL CONSGR$
COL HEAD 2 = $/CL POE$
COL HEAD 3 = $/CL POD$
COL HEAD 4 = $TOT WGT/(TONS)$
COL HEAD 5 = $/TOT CUBE$
COL HEAD 6 = $/MTONS$
COL HEAD 7 = $/TOT-CHARGE$
COL HEAD 8 = $LH RATE/PER LB$
PRINT SHIP-PT, POE-ADR, POD-ADR, WGT-TONS,
CUBE, MT-CUBE, TOT-CHARGE, LHRATE
$END
```

The data elements defined in this report include: consignor city and state, in the clear POE and DoD addresses, total cargo weight, total cargo cube, cargo measurement tons, total paid charges, and the linehaul rate (dollars/pound).

Figure 7 - Seavan Cost Report

```
ID CONDA
REPORT R03
WIDTH = 135
MHEAD = $SEAVAN CONTENT ROUTE DATA$
COLHEAD 1 = $CONSIGNOR/ADDRESS$
COLHEAD 2 = $POE/ADDRESS$
COLHEAD 3 = $POD/ADDRESS$
COLHEAD 4 = $COMMODITY/CODE$
COLHEAD 5 = $TCN/NUMBER$
COLHEAD 6 = $WEIGHT/(LBS)$
COLHEAD 7 = $CUBE/(CU FT)$
COLHEAD 8 = $DATE REC/AT POE$
PRINT CONSGR-ADR,POE-ADR,POD-ADR,COM,TCN-NO,WEIGHT,CUBE, AND DTE-REC
$END
```

Report R03 contains these data elements; consignor address, POE address, POD address, commodity code, transportation control number, weight, cube, and receipt date of cargo at the POE.

Figure 8 - Seavan Content Route Report

```
ID CONDA
REPORT R04
WIDTH = 135
MHEAD = $INLAND COST SHIPMENT DATA$
COLHEAD 1 = $/SHIPPING/POINT$
COLHEAD 2 = $/DESTINATION/POINT$
COLHEAD 3 = $TOTAL/WEIGHT/(LBS)$
COLHEAD 4 = $TOTAL/CUBE/(CU FT)$
COLHEAD 5 = $TOTALCHARGES/(DOL&CTS)$
COLHEAD 6 = $LINEHAUL/RATE$
COLHEAD 7 = $DELIVERY/DATE/TO DEST PT$
PRINT SHIP-PT,DEST-PT,TOT-WGT,TOT-CUBE,TOT-CHARGE,LHRATE,DLVY-DATE
$END
```

Report R04 is a summary of inland shipment cost data and includes these data elements: the in the clear or noncoded shipping point, destination point, total shipment weight, total cube, total paid charges, linehaul rate, and the delivery date to the destination point.

Figure 9 - Inland Cost Shipment Report

```
ID CONDA
REPORT R05
WIDTH = 135
MHEAD = $NAVY PORT HANDLING CHARGES$
COLHEAD 1  = $/POE$
COLHEAD 2  = $/POD$
COLHEAD 3  = $TYPE OF/MOVE$
COLHEAD 4  = $COMMODITY/CODE$
COLHEAD 5  = $TRANSSHIPPED/RATE$
COLHEAD 6  = $TRANSSHIPPED/MTONS$
COLHEAD 7  = $PREMIUM/RATE$
COLHEAD 8  = $PREMIUM/MTONS$
COLHEAD 10 = $PREMIUM/CHARGES$
COLHEAD 11 = $BILLING/MONTH$
PRINT POE2, POD2, TYPE-MOV2, CONT-CODE, TRANS-RATE, TRANS-MTON,
PREM-RATE, PREM-MTON, THRU-CHGE, PREM-CHGE, BILL-MO
$END
```

The data elements which compose report R05 are: POE, POD, type of movement (either import or export), commodity code, transshipment rate, transshipment measurement tons, premium rates, premium measurement tons, transshipped charges, premium charges, and the billing month.

Figure 10 - Port Handling Charge Report

```
ID CONDA
REPORT RO6
WIDTH = 135
MHEAD = $NAVY SEALIFT CHARGES$
COLHEAD 1  = $/POE$
COLHEAD 2  = $/POD$
COLHEAD 3  = $INITIAL/ORIGIN$
COLHEAD 4  = $ULTIMATE/DESTINATION$
COLHEAD 5  = $/COMMODITY$
COLHEAD 6  = $QUANTITY/(MTONS)$
COLHEAD 7  = $COST/(DOLLARS)$
COLHEAD 8  = $SEALIFT RATE$
COLHEAD 9  = $SHIPMENT/DIRECTION$
COLHEAD 10 = $ESTIMATED/PAYMENT$
COLHEAD 11 = $REPORT/DATE$
PRINT POE3, POD3, ORG-MSC, DEST-MSC, COM-MSC, WGT-MSC, COST-MSC,
SEARATE, SHIP-DIR, ESTIMATE, RDATE-MSC
$END
```

Report RO6 reflects the port-to-port segment of the transportation network. Data elements include the POE, POD, origin point, destination point, commodity code, weight, cost of shipment, sealift rate, direction of shipment (import or export), estimated charges indicator, and receipt date.

Figure 11 - Sealift Charge Report

```
ID CONDA
REPORT R07
WIDTH = 72
MHEAD = $INLAND COST SHIPMENT DATA$
COLHEAD 1 = $/SHIPPING POINT$
COLHEAD 2 = $/DESTINATION/POINT$
COLHEAD 3 = $TOTAL/WEIGHT/(LBS)$
COLHEAD 4 = $TOTAL/CUBE/(CU FT)$
COLHEAD 5 = $TOTAL CHARGES/(DOL&CTS)
COLHEAD 6 = $LINEHAUL/RATE$
COLHEAD 7 = $DELIVERY/DATE/TO DEST PT$
PRINT SHIP-PT, DEST-PT, TOT-WGT, TOT-CUBE, TOT-CHARGE, LHRATE, DLVY-DATE
$END
```

Report R07 is identical to the Inland Cost Shipment Report R04 except that it is defined a 72-column rather than a 135-column report.

Figure 12 - Inland Shipment Report

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Mr. Raymond Melton for his efforts in preparing the initial data files for CONDA. The authors are also indebted to Mr. Ben Wallis and Mr. Clarence Perry for their consultation and support in the design and development of the data base.

REFERENCES

1. "Container Oriented Network Distribution Analysis," J.J. Henry Co., Inc., Portsmouth, Virginia (Oct 1978).

2. Wallis, Ben et al., "The SHARP Data Base Management System User Information Manual," Report TM-188-76-1, First Revision (Aug 1976).

3. "Military Standard Transportation and Movement Procedures," DoD Regulation 4500.32-R, Volumes I and II (1 Jan 1975).

INITIAL DISTRIBUTION

Copies	
10	NAVMTO
10	MSC
25	MTMC
	10 EA
	10 WA
	5 HQS
5	FMSO
5	MAC
18	NAVSUP
	5 SUP 043
	10 SUP 05
	3 SUP
5	NSC Norfolk
5	NSC Oakland
5	NSC Charleston
5	NSC Pudget Sound
5	NSC Pearl Harbor
5	NSC San Diego
12	DTIC

CENTER DISTRIBUTION

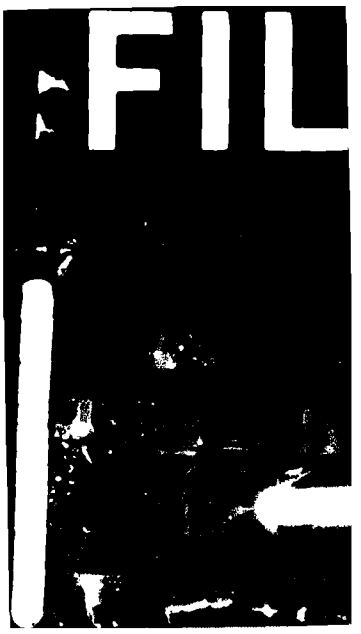
Copies	Code	Name
1	18	G.H. Gleissner
2	1809.3	D. Harris
1	187	M. Zubkoff
50	187	F. Myers
10	5211.1	Reports Distribution
1	522.1	Unclassified Lib (C)
1	522.2	Unclassified Lib (A)

DTNSRDC ISSUES THREE TYPES OF REPORTS

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.

2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.

3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.


FIL